

 **PORTAL**  
USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login  
**Search:**  The ACM Digital Library  The Guide  
 +memory +stack, +program +counter, +interrupt, +push, +pop

THE ACM DIGITAL LIBRARY

 [Feedback](#) [Report a problem](#) [Satisfaction survey](#)

## Terms used

[memory](#) [stack](#) [program](#) [counter](#) [interrupt](#) [push](#) [pop](#) [bank](#) [processor](#) [architecture](#)

Found 34 of 167,655

Sort results by



 Save results to a Binder[Try an Advanced Search](#)

Display results



 [Search Tips](#)[Try this search in The ACM Guide](#) Open results in a new window

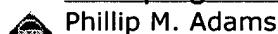
Results 1 - 20 of 34

Result page: [1](#) [2](#) [next](#)Relevance scale **1 Static checking of interrupt-driven software**

Dennis Brylow, Niels Damgaard, Jens Palsberg

July 2001 **Proceedings of the 23rd International Conference on Software Engineering****Publisher:** IEEE Computer SocietyFull text available:  [pdf\(157.76 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)  
 [Publisher Site](#)

Resource-constrained devices are becoming ubiquitous. Examples include cell phones, palm pilots, and digital thermostats. It can be difficult to fit required functionality into such a device without sacrificing the simplicity and clarity of the software. Increasingly complex embedded systems require extensive brute-force testing, making development and maintenance costly. This is particularly true for system components that are written in assembly language. Static checking has the potential o ...

**2 Microprogrammable microprocessor survey**June 1978 **ACM SIGMICRO Newsletter**, Volume 9 Issue 2**Publisher:** ACM PressFull text available:  [pdf\(1.24 MB\)](#) Additional Information: [full citation](#), [abstract](#)

The Motorola M10800 LSI processor family consists of a sequencer, referred to as a Microprogram Control Function (MCF) - MC10801, and a processing element, referred to as a 4-bit ALU Slice - MC10800 (Not to be confused with the processor family number M10800). Undoubtedly, the most interesting feature of the M10800 processor family is the ECL technology used to produce it. The M10800 processor family is completely MECL 10,000 compatible and exhibits the ultra-high speed performance of ECL logic.

**3 μ3L: An HLL-RISC processor for parallel execution of FP-language programs**

M. Castan, E. I. Organick

April 1982 **Proceedings of the 9th annual symposium on Computer Architecture****Publisher:** IEEE Computer Society PressFull text available:  [pdf\(709.88 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

To eliminate the conceptual distance between the hardware instruction set and the user interface, some architects advocate High Level Language (HLL) machines. To obtain simple, fast and cheap machines, some architects advocate Reduced Instruction Set

Computer (RISC) machines. This paper reconciles both views and presents an architecture which has both an HLL user interface and a RISC hardware. Each instance of this architecture is a module of an HLL multiprocessor system. Function ...

#### 4 Chap - a SIMD graphics processor

 Adam Levinthal, Thomas Porter

January 1984 **ACM SIGGRAPH Computer Graphics , Proceedings of the 11th annual conference on Computer graphics and interactive techniques SIGGRAPH '84**, Volume 18 Issue 3

Publisher: ACM Press

Full text available:  pdf(533.22 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

Special purpose processing systems designed for specific applications can provide extremely high performance at moderate cost. One such processor is presented for executing graphics and image processing algorithms as the basis of a digital film printer. Pixels in the system contain four parallel components: RGB for full color and an alpha channel for retaining transparency information. The data path of the processor contains four arithmetic elements connected through a crossbar network ...

**Keywords:** Compositing, Computer graphics, Digital film printers, Parallel processing, SIMD architecture, Tesselation

#### 5 A RISC architecture for symbolic computation

 Richard B. Kieburtz

October 1987 **ACM SIGARCH Computer Architecture News , ACM SIGPLAN Notices , ACM SIGOPS Operating Systems Review , Proceedings of the second international conference on Architectural support for programming languages and operating systems ASPLOS-II**, Volume 15 , 22 , 21 Issue 5 , 10 , 4

Publisher: IEEE Computer Society Press, ACM Press

Full text available:  pdf(1.20 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The G-machine is a language-directed processor architecture designed to support graph reduction as a model of computation. It can carry out lazy evaluation of functional language programs and can evaluate programs in which logical variables are used. To support these language features, the abstract machine requires tagged memory and executes some rather complex instructions, such as to evaluate a function application. This paper explores an implementation of the G-machine as a high performance RI ...

#### 6 Swamp: a fast processor for Smalltalk-80

 David M. Lewis, David R. Galloway, Robert J. Francis, Brian W. Thomson

June 1986 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications OOPSLA '86**, Volume 21 Issue 11

Publisher: ACM Press

Full text available:  pdf(849.10 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A processor for the Smalltalk-80<sup>†</sup> programming language is described. This machine is implemented using a standard bit slice ALU and sequencer, TTL MSI, and NMOS LSI RAMS. It executes an instruction set similar to the Smalltalk-80 virtual machine instruction set. The data paths of the machine are optimized for rapid Smalltalk-80 execution by the inclusion of a context cache, tag checking, and a hardware method cache. Each context is only partly initialized when created, and has no memory ...

8 Special issue: Game-playing programs: theory and practice

M. A. Bramer

April 1982 **ACM SIGART Bulletin**, Issue 80

Publisher: ACM Press

Full text available: [pdf\(9.23 MB\)](#) Additional Information: [full citation](#), [abstract](#)

This collection of articles has been brought together to provide SIGART members with an overview of Artificial Intelligence approaches to constructing game-playing programs. Papers on both theory and practice are included.

8 Sh-BOOM: the sound of the RISC market changing

George William Shaw

March 1991 **Proceedings of the second and third annual workshops on Forth**

Publisher: ACM Press

Full text available: [pdf\(686.65 KB\)](#) Additional Information: [full citation](#), [index terms](#)

9 Evolving minicomputer architecture

T. G. Lewis

October 1977 **ACM SIGMINI Newsletter**, Volume 3 Issue 4

Publisher: ACM Press

Full text available: [pdf\(951.85 KB\)](#) Additional Information: [full citation](#), [references](#)

10 Register allocation for free: The C machine stack cache

David R. Ditzel, H. R. McLellan

March 1982 **ACM SIGPLAN Notices , ACM SIGARCH Computer Architecture News , Proceedings of the first international symposium on Architectural support for programming languages and operating systems ASPLOS-I**, Volume 17 , 10 Issue 4 , 2

Publisher: ACM Press

Full text available: [pdf\(673.92 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

The Bell Labs C Machine project is investigating computer architectures to support the C programming language.<sup>1</sup> One of the goals is to match an efficient architecture to the language and the compiler technology available. Measurements of different C programs show that roughly one out of every twenty instructions executed is either a procedure call or return.<sup>2</sup> Procedure call overhead is therefore a very important consideration in the overall machine ...

11 The microarchitecture of a capability-based computer

D. A. Abramson, J. Rosenberg

December 1986 **ACM SIGMICRO Newsletter , Proceedings of the 19th annual workshop on Microprogramming MICRO 19**, Volume 17 Issue 4

Publisher: ACM Press

Full text available: [pdf\(831.64 KB\)](#) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

This paper describes the micro-architecture of a microprogrammed workstation called MONADS-PC. The system has been specifically designed to support a very large uniform virtual memory, capability-based addressing and information hiding software modules with procedural interfaces. The paper gives a brief introduction to these topics followed by implementation details of the system.

 Measuring VAX 8800 performance with a histogram hardware monitor

D. W. Clark, P. J. Bannon, J. B. Keller

May 1988 **ACM SIGARCH Computer Architecture News , Proceedings of the 15th Annual International Symposium on Computer architecture ISCA '88,**

Volume 16 Issue 2

Publisher: IEEE Computer Society Press, ACM Press

Full text available:  pdf(1.04 MB)

Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper reports the results of a study of VAX 8800 processor performance using a hardware monitor that collects histograms of the processor's micro-PC and memory bus status. The monitor keeps a count of all machine cycles executed at each micro-PC location, as well as counting all occurrences of each bus transaction. It can measure a running system without interfering with it, and this paper's results are based on measurements of live timesharing. Because the 8800 is a microcoded machine ...

**13 The micro-architecture of the ECLIPSE® MV/8000: Conception and implementation**

 Jonathan S. Blau, Charles J. Holland, David L. Keating

November 1980 **ACM SIGMICRO Newsletter , Proceedings of the 13th annual workshop on Microprogramming MICRO 13**, Volume 11 Issue 3-4

Publisher: IEEE Press, ACM Press

Full text available:  pdf(622.95 KB)

Additional Information: [full citation](#), [abstract](#), [index terms](#)

The microcode of the ECLIPSE MV/8000 controls the hardware to emulate an instruction set. In the MV/8000 the micro-architecture is defined and limited by the following constraints: 1) the desire to implement microcode in a limited number of locations; 2) the use of LSI technology; 3) a virtual memory architecture. This paper will attempt to show how each of these factors contributed to the micro-architecture, to describe that architecture, and to relat ...

**14 Self-assessment procedure XII: a self-assessment procedure dealing with computer architecture**

 Robert I. Winner, Edward M. Carter

January 1984 **Communications of the ACM**, Volume 27 Issue 1

Publisher: ACM Press

Full text available:  pdf(589.25 KB)

Additional Information: [full citation](#), [references](#), [index terms](#)

**15 Burroughs Corporation: corporate public relations**

 October 1974 **ACM SIGMICRO Newsletter**, Volume 5 Issue 3

Publisher: ACM Press

Full text available:  pdf(7.37 MB)

Additional Information: [full citation](#)

**16 The Postroom Computer**

 Hugh Osborne

December 2001 **Journal on Educational Resources in Computing (JERIC)**, Volume 1 Issue 4

Publisher: ACM Press

Full text available:  pdf(242.80 KB)

Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

The Postroom Computer is a computer architecture simulator based on the *Little Man Computer* developed in 1965 by Stuart Madnick and John Donovan. It provides a family of architectures suitable for use in teaching introductory computer architectures. It is designed to introduce aspects of computer architecture and low-level programming in an incremental way. The extensions are designed to provide a range of computing models

within the Little Man Computer paradigm. As they are introduced th ...

**Keywords:** Computer architecture simulator, education

**17 The reduction of branch instruction execution overhead using structured control flow** 

 Robert G. Wedig, Marc A. Rose  
January 1984 **ACM SIGARCH Computer Architecture News , Proceedings of the 11th annual international symposium on Computer architecture ISCA '84,**  
Volume 12 Issue 3

**Publisher:** ACM Press

Full text available:  pdf(707.36 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

This paper presents a technique for specifying change of control (e.g. branch) commands at a sequential processor's macroinstruction set level. It is shown that by representing high level language (HLL) control statements with special machine language instructions, the usual delays associated with control flow changes can be reduced. Preserving the HLL control flow information increases performance by reducing both the number of executed branches and pipeline breaks.

**18 List processing in real time on a serial computer** 

 Henry G. Baker  
April 1978 **Communications of the ACM**, Volume 21 Issue 4

**Publisher:** ACM Press

Full text available:  pdf(1.55 MB) Additional Information: [full citation](#), [abstract](#), [references](#), [citations](#), [index terms](#)

A real-time list processing system is one in which the time required by the elementary list operations (e.g. CONS, CAR, CDR, RPLACA, RPLACD, EQ, and ATOM in LISP) is bounded by a (small) constant. Classical implementations of list processing systems lack this property because allocating a list cell from the heap may cause a garbage collection, which process requires time proportional to the heap size to finish. A real-time list processing system is presented which continuously reclaims garb ...

**Keywords:** CDR-coding, LISP, compacting, file or database management, garbage collection, list processing, real-time, reference counting, storage allocation, storage management, virtual memory

**19 Communication systems: A second-generation sensor network processor with** 

 application-driven memory optimizations and out-of-order execution

Leyla Nazhandali, Michael Minuth, Bo Zhai, Javin Olson, Todd Austin, David Blaauw  
September 2005 **Proceedings of the 2005 international conference on Compilers, architectures and synthesis for embedded systems CASES '05**

**Publisher:** ACM Press

Full text available:  pdf(183.09 KB) Additional Information: [full citation](#), [abstract](#), [references](#), [index terms](#)

In this paper we present a second-generation sensor network processor which consumes less than one picoJoule per instruction (typical processors use 100's to 1000's of picoJoules per instruction). As in our first-generation design effort, we strive to build microarchitectures that minimize area to reduce leakage, maximize transistor utility to reduce the energy-optimal voltage, and optimize CPI for efficient processing. The new design builds on our previous work to develop a low-power subthresho ...

**Keywords:** energy efficiency, memory organization, microprocessor, sensor network

20 [FLIP-FLOP: a stack-oriented multiprocessing system](#)



 Peter Grabienki

March 1991 **ACM SIGARCH Computer Architecture News**, Volume 19 Issue 1

Publisher: ACM Press

Full text available:  [pdf\(672.87 KB\)](#) Additional Information: [full citation](#), [abstract](#), [index terms](#)

Many research and application fields are today computationally very demanding. This requirement has influenced the development of high-performance processors, vector processors and multiprocessor engines. This paper describes the development of the multiprocessing system FLIP-FLOP (Fast Link Periphery for a Forth Language Oriented Processor) which combines a stack oriented processor kernel and a communication coprocessor for message passing. In the past many multiprocessor systems based on availab ...

Results 1 - 20 of 34

Result page: [1](#) [2](#) [next](#)

The ACM Portal is published by the Association for Computing Machinery. Copyright © 2005 ACM, Inc.

[Terms of Usage](#) [Privacy Policy](#) [Code of Ethics](#) [Contact Us](#)

Useful downloads:  [Adobe Acrobat](#)  [QuickTime](#)  [Windows Media Player](#)  [Real Player](#)

[Home](#) | [Login](#) | [Logout](#) | [Access Information](#) | [About](#)

Welcome United States Patent and Trademark Office

[Search Session History](#)[BROWSE](#)[SEARCH](#)[IEEE XPLORE GUIDE](#)

Wed, 23 Nov 2005, 12:03:30 PM EST

Edit an existing query or compose a new query in the Search Query Display.

[Search Query Display](#)**Select a search number (#) to:**

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

[Recent Search Queries](#)

#1	(( memory stack<in>metadata ) <and> ( stack<in>metadata ) )<and> ( interrupt<in>metadata )
#2	(program counter<IN>metadata)
#3	(memory bank<IN>metadata)
#4	(data stack<IN>metadata)
#5	(condition code register<IN>metadata)
#6	1 and 2

[Help](#) [Contact Us](#) [Privacy](#)

© Copyright 2005 IE

Indexed by  
**Inspec**

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	9	(Santi near Carlo).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:27
L2	5	(Edmondo near Gangi).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:27
L3	2	1 and 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:41
L4	12	1 or 2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:27
L5	2018	memory adj stack	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:41
L6	1903	data adj stack	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:42
L7	15849	program adj counter	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:42
L8	811	condition adj code adj register	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:44
L9	636548	interrupt\$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:44
L10	7116	(push\$3 or pop\$3) adj operation	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:46

L11	18273	memory near2 bank\$2	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:46
L12	974	9 and 10	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:46
L13	20	12 and 8	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:56
L15	11	7 and 13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:48
L16	2	5 and 15	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:48
L17	1	4 and 13	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2005/11/23 11:56